Informal Argonne Seminar Series, 2004

The Revival of Active Set Methods

Sven Leyffer, leyffer@mcs.anl.gov

Mathematics & Computer Science Division, Argonne National Laboratory

1. Why Do We Need Active Set Methods?

2. Active Set Methods for Quadratic Programs

3. Active Set Methods for Nonlinear Programs

Why Do We Need Active Set Methods (ASMs)?

Philosophy
Lesson

Issues in Optimization Solvers

$$\underset{x}{\text{minimize}}\; f(x) \qquad \text{subject to}\; c(x) \geq 0$$

1. **Global Convergence**
2. **Active Set Identification**
3. **Fast Local Convergence**

Issues in Optimization Solvers

$$\underset{x}{\text{minimize}}\ f(x) \qquad \text{subject to}\ c(x) \geq 0$$

1. **Global Convergence**
   - merit function, e.g. $f(x) + \pi \| c(x)^- \|$ for $\pi > \| y^* \|_D$
   - filter ... more later
2. **Active Set Identification**
3. **Fast Local Convergence**

Issues in Optimization Solvers

$$\underset{x}{\text{minimize}}\ f(x) \qquad \text{subject to}\ c(x) \geq 0$$

1. **Global Convergence**
   - merit function, e.g. $f(x) + \pi \| c(x)^- \|$ for $\pi > \| y^* \|_D$
   - filter ... more later
2. **Active Set Identification**
   - given active set, simply use Newton's method
   - step computation: update estimate of active set
   - alternative: interior point methods $Yc(x) = \mu e$ & $\mu \searrow 0$
3. **Fast Local Convergence**

Issues in Optimization Solvers

$$\underset{x}{\text{minimize}}\ f(x) \qquad \text{subject to } c(x) \geq 0$$

1. **Global Convergence**
   - merit function, e.g. $f(x) + \pi \|c(x)^-\|$ for $\pi > \|y^*\|_D$
   - filter ... more later
2. **Active Set Identification**
   - given active set, simply use Newton's method
   - step computation: update estimate of active set
   - alternative: interior point methods $Yc(x) = \mu e$ & $\mu \searrow 0$
3. **Fast Local Convergence**
   - conjugate gradients et al.
   - (constrained) preconditioners

Why Do We Need Active Set Methods (ASMs)?

Interior point methods (IPMs) usually faster than ASMs:

1. Pivoting inefficient for huge problems
2. Single QP solve $\simeq$ several Newton steps of IPM
3. Null-space projected Hessian factors are **DENSE**

Why Do We Need Active Set Methods (ASMs)?

Interior point methods (IPMs) usually faster than ASMs:

1. Pivoting inefficient for huge problems
2. Single QP solve $\simeq$ several Newton steps of IPM
3. Null-space projected Hessian factors are **DENSE**

Why should we be interested in ASMs?

- ASMs often more robust than interior point methods
- ASMs better for warm-starts (repeated solves)
- Easier to precondition ... iterative solves

Challenge: overcome 1. & 2. from above

Why Do We Need Active Set Methods (ASMs)?

Interior point methods (IPMs) usually faster than ASMs:

1. Pivoting inefficient for huge problems
2. Single QP solve $\simeq$ several Newton steps of IPM
3. Null-space projected Hessian factors are **DENSE**

Why should we be interested in ASMs?

- ASMs often more robust than interior point methods
- ASMs better for warm-starts (repeated solves)
- Easier to precondition ... iterative solves

Challenge: overcome 1. & 2. from above

Two ways to make large changes to active set

1. Projected gradient approach
2. Sequential linear programming approach

ACTIVE SET METHODS FOR QPs

Active Sets for Quadratic Programs (QPs)

$$\boxed{\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}x^T H x + g^T x \\ \text{subject to} & A^T x = b \\ & l \le x \le u \end{array}}$$

- H is symmetric (indefinite?)
- A^T is $m \times n$, $m < n$, full rank
- General: $\bar{l} \le \begin{pmatrix} x \\ A^T x \end{pmatrix} \le \bar{u}$

$$\begin{array}{ll} \text{Active set} & \mathcal{A}(x) = \{i \mid x_i = l_i \text{ or } x_i = u_i\} \\ \text{Inactive set} & \mathcal{I}(x) = \{1, \ldots, n\} - \mathcal{A}(x) \end{array}$$

Active Sets for Quadratic Programs (QPs)

$$\begin{array}{ll} \text{Active set} & \mathcal{A}(x) = \{i \mid x_i = l_i \text{ or } x_i = u_i\} \\ \text{Inactive set} & \mathcal{I}(x) = \{1, \ldots, n\} - \mathcal{A}(x) \end{array}$$

Given $\mathcal{A}$, QP solution $(x^*_{\mathcal{I}}, y^*)$ solves

$$\begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}} \\ A^T_{:,\mathcal{I}} & \end{bmatrix} \begin{pmatrix} x_{\mathcal{I}} \\ y \end{pmatrix} = \begin{pmatrix} -g_{\mathcal{I}} - H_{\mathcal{I},\mathcal{A}} x_{\mathcal{A}} \\ b - A^T_{:,\mathcal{A}} x_{\mathcal{A}} \end{pmatrix}$$

Active-set methods search for $\mathcal{A}^*$:

- Delete entries from $\mathcal{A}^k$; update a factorization; compute step
- Possibly add entries to $\mathcal{A}^k$
- $\exists$ robust solvers; good for warm starts ... n large ???

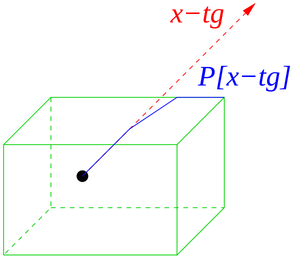
PROJECTED GRADIENT

Projected Gradient for Box Constrained QPs

Simpler box constrained QP ...

$$\left\{ \begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}x^THx + g^Tx \; =: \; q(x) \\ \text{subject to} & l \le x \le u \end{array} \right.$$

Projected steepest descent $P[x - \alpha \nabla q(x)]$
○ piecewise linear path
... large changes to $\mathcal{A}$-set
... but slow (steepest descent)



x^c Cauchy point $\equiv$ first minimum of $q(x(\alpha))$, for $\alpha \ge 0$

Theorem: Cauchy points converge to stationary point.

Projected Gradient & CG for Box Constrained QPs

```
$x^0$ given such that $l \le x^0 \le u$; set $k = 0$
WHILE (not optimal) BEGIN
    1. find Cauchy point $x_k^c$ & active set $\mathcal{A}(x_k^c)$
    2. (approx.) solve box QP in subspace $\mathcal{I} := \{1, \ldots, n\} - \mathcal{A}(x_k^c)$
       $\boxed{\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}x^T Hx + g^T x \\ \text{subject to} & l \le x \le u \\ & x_i = [x_k^c]_i \; , \forall \; i \in \mathcal{A}(x_k^c) \end{array}} \Leftrightarrow \boxed{\begin{array}{l} \text{apply CG to ...} \\ H_{\mathcal{I},\mathcal{I}} x_{\mathcal{I}} = \ldots \end{array}}$
       for $x^{k+1}$; set $k = k + 1$
END
```

Cauchy point $\Rightarrow$ global convergence ... but faster due to CG

How to Include $A^T x = b$?

Projection onto box is easy, but tough for general QP

$$P_{QP}[z] = \left\{ \begin{array}{ll} \underset{x}{\text{minimize}} & (x-z)^T(x-z) \\ \text{subject to} & A^T x = b \\ & l \leq x \leq u \end{array} \right.$$

... as hard as original QP! ... Idea: project onto box only

$\Rightarrow$ subspace solve $H_{\mathcal{I},\mathcal{I}} x_{\mathcal{I}} = \dots$ becomes solve with KKT system

$$\begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}} \\ A^T_{:,\mathcal{I}} & \end{bmatrix} \begin{pmatrix} x_{\mathcal{I}} \\ y \end{pmatrix} = \dots$$

Which gradient / merit function in Cauchy step?

AUGMENTED LAGRANGIAN

The Augmented Lagrangian

Arrow & Solow ('58), **Hestenes** ('69), **Powell** ('69)

$$\boxed{\underset{x}{\text{minimize}}\ L(x, y_k, \rho_k) \ = \ f(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k \|c(x)\|^2}$$

As $y_k \to y_*$:
- $x_k \to x_*$ for $\rho_k > \bar{\rho}$
- No ill-conditioning, improves convergence rate

- An old idea for nonlinear constraints ... smooth merit function
- Poor experience with LPs (e.g., MINOS vs. LANCELOT)
- But special structure of LPs (and QPs) not fully exploited

$$\boxed{f(x) = \tfrac{1}{2}x^T Hx + g^T x \quad \& \quad c(x) = A^T x - b}$$

Bound Constrained Lagrangian (BCL)

Minimizing the augmented Lagrangian subject to bounds:

WHILE (not optimal) **BEGIN**

1. Find $\omega_k \searrow 0$ optimal x_k of

$$\boxed{\underset{l \le x \le u}{\text{minimize}} \quad f(x) - y_k^T c(x) + \frac{1}{2}\rho_k \|c(x)\|^2}$$

Bound Constrained Lagrangian (BCL)

Minimizing the augmented Lagrangian subject to bounds:

```
WHILE (not optimal) BEGIN
  1. Find $\omega_k \searrow 0$ optimal $x_k$ of
     $\boxed{\underset{l \le x \le u}{\text{minimize}} \quad f(x) - y_k^T c(x) + \frac{1}{2}\rho_k \|c(x)\|^2}$
  2. IF $\|c(x_k)\| \le \eta_k \searrow 0$ THEN
         Update $y_k$ ( typically $y_{k+1} = y_k - \rho_k c(x_k)$ )
     ELSE increase $\rho_k$
END
```

Bound Constrained Lagrangian (BCL)

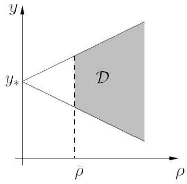
Minimizing the augmented Lagrangian subject to bounds:

```
**WHILE** (not optimal) **BEGIN**
    1. Find $\omega_k \searrow 0$ optimal $x_k$ of
       $\boxed{\underset{l \le x \le u}{\text{minimize}} \quad f(x) - y_k^T c(x) + \frac{1}{2}\rho_k \|c(x)\|^2}$
    2. **IF** $\|c(x_k)\| \le \eta_k \searrow 0$ **THEN**
           Update $y_k$ ( typically $y_{k+1} = y_k - \rho_k c(x_k)$ )
       **ELSE** increase $\rho_k$
**END**
```

Arbitrary sequences: $\eta_k \& \omega_k$ control feasibility & optimality

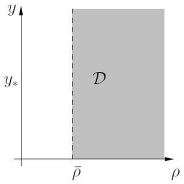
Augmented Lagrangian for Linear Constraints

Nonlinear $c(x)$



$(y_k, \rho_k) \in \mathcal{D}$

$c(x) = A^T x - b$



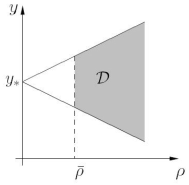
$\rho_k > \bar{\rho}$

$\forall (\rho, y) \in \mathcal{D}$, minimize $L(x, y, \rho)$ has unique solution $x(y, \rho)$:

- bound constrained augmented Lagrangian converges
- Hessian $\nabla^2_{xx} L(x, y, \rho)$ is positive definite on optimal face

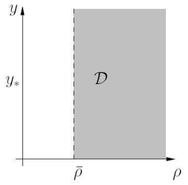
Augmented Lagrangian for Linear Constraints

Nonlinear $c(x)$



$(y_k, \rho_k) \in \mathcal{D}$

$c(x) = A^T x - b$



$\rho_k > \bar\rho$

$\forall(\rho, y) \in \mathcal{D}$, minimize $L(x, y, \rho)$ has unique solution $x(y, \rho)$:
- bound constrained augmented Lagrangian converges
- Hessian $\nabla^2_{xx} L(x, y, \rho)$ is positive definite on optimal face

$$\boxed{\bar\rho \approx 2 \frac{\|H_*\|}{\|A_* A^T{}_*\|}}$$ ... depends on active set ... from dual Hessian

QP by Projected Augmented Lagrangian QPPAL

```
WHILE (not optimal) BEGIN
    1. Find $\omega_k \searrow 0$ optimal solution $x_k^c$ of
       $\boxed{\underset{l \le x \le u}{\text{minimize}} \; \frac{1}{2}x^T Hx + g^T x - y^T(A^T x - b) + \frac{1}{2}\rho_k \|A^T x - b\|^2}$
    2. Find $\mathcal{A}(x_k^c)$ & estimate penalty $\bar{\rho} = 2\,\|H_{\mathcal{I}}\|/\|A_{\mathcal{I}} A_{\mathcal{I}}^T\|$
    3. IF $\bar{\rho} > \rho_k$ THEN update $\rho_{k+1} = \bar{\rho}$ & CYCLE
       ELSE update multiplier: $y_k^c = y_k - \rho_k(A^T x_k^c - b)$
```

QP by Projected Augmented Lagrangian QPPAL

WHILE (not optimal) **BEGIN**

1. Find $\omega_k \searrow 0$ optimal solution x_k^c of
$$\boxed{\underset{l \le x \le u}{\text{minimize}} \ \tfrac{1}{2}x^THx + g^Tx - y^T(A^Tx - b) + \tfrac{1}{2}\rho_k\|A^Tx - b\|^2}$$
2. Find $\mathcal{A}(x_k^c)$ & estimate penalty $\bar{\rho} = 2\,\|H_{\mathcal{I}}\|/\|A_{\mathcal{I}}A_{\mathcal{I}}^T\|$
3. **IF** $\bar{\rho} > \rho_k$ **THEN** update $\rho_{k+1} = \bar{\rho}$ & **CYCLE**
   ELSE update multiplier: $y_k^c = y_k - \rho_k(A^Tx_k^c - b)$
4. Solve equality QP in subspace $\longrightarrow \ (\Delta_{x_{\mathcal{I}}}, \Delta y)$
$$\begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}} \\ A_{:,\mathcal{I}}^T & \end{bmatrix} \begin{pmatrix} \Delta x_{\mathcal{I}} \\ \Delta y \end{pmatrix} = -\begin{pmatrix} [\nabla x L(x_k^c, y_k^c, 0)]_{\mathcal{I}} \\ A^Tx_k^c - b \end{pmatrix}$$

QP by Projected Augmented Lagrangian QPPAL

```
WHILE (not optimal) BEGIN
    1. Find $\omega_k \searrow 0$ optimal solution $x_k^c$ of
       $\boxed{\underset{l \le x \le u}{\text{minimize}} \ \frac{1}{2}x^T Hx + g^T x - y^T(A^T x - b) + \frac{1}{2}\rho_k \|A^T x - b\|^2}$
    2. Find $\mathcal{A}(x_k^c)$ & estimate penalty $\bar{\rho} = 2\,\|H_{\mathcal{I}}\| / \|A_{\mathcal{I}} A_{\mathcal{I}}^T\|$
    3. IF $\bar{\rho} > \rho_k$ THEN update $\rho_{k+1} = \bar{\rho}$ & CYCLE
       ELSE update multiplier: $y_k^c = y_k - \rho_k(A^T x_k^c - b)$
    4. Solve equality QP in subspace $\rightarrow \ (\Delta_{x_{\mathcal{I}}}, \Delta y)$
       $\begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}} \\ A_{:,\mathcal{I}}^T & \end{bmatrix} \begin{pmatrix} \Delta x_{\mathcal{I}} \\ \Delta y \end{pmatrix} = -\begin{pmatrix} [\nabla x L(x_k^c, y_k^c, 0)]_{\mathcal{I}} \\ A^T x_k^c - b \end{pmatrix}$
    5. Line-search on $L(x_k^c + \alpha\Delta x, y_k^c + \alpha\Delta y, \rho)$; update $x, y, k, \rho$
END
```

QP by Projected Augmented Lagrangian QPPAL

```
WHILE (not optimal) BEGIN
    1. Find $\omega_k \searrow 0$ optimal solution $x_k^c$ of
       $\boxed{\underset{l \le x \le u}{\text{minimize}} \ \frac{1}{2} x^T H x + g^T x - y^T (A^T x - b) + \frac{1}{2} \rho_k \|A^T x - b\|^2}$
    2. Find $\mathcal{A}(x_k^c)$ & estimate penalty $\bar{\rho} = 2 \, \|H_{\mathcal{I}}\| / \|A_{\mathcal{I}} A_{\mathcal{I}}^T\|$
    3. IF $\bar{\rho} > \rho_k$ THEN update $\rho_{k+1} = \bar{\rho}$ & CYCLE
       ELSE update multiplier: $y_k^c = y_k - \rho_k (A^T x_k^c - b)$
    4. Solve equality QP in subspace $\longrightarrow \ (\Delta_{x_{\mathcal{I}}}, \Delta y)$
       $\begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}} \\ A_{:,\mathcal{I}}^T & \end{bmatrix} \begin{pmatrix} \Delta x_{\mathcal{I}} \\ \Delta y \end{pmatrix} = - \begin{pmatrix} [\nabla x L(x_k^c, y_k^c, 0)]_{\mathcal{I}} \\ A^T x_k^c - b \end{pmatrix}$
    5. Line-search on $L(x_k^c + \alpha \Delta x, y_k^c + \alpha \Delta y, \rho)$; update $x, y, k, \rho$
END
```

1.-3. identify active set

QP by Projected Augmented Lagrangian QPPAL

```
WHILE (not optimal) BEGIN
    1. Find $\omega_k \searrow 0$ optimal solution $x_k^c$ of
       $\boxed{\underset{l \le x \le u}{\text{minimize}} \; \frac{1}{2}x^T Hx + g^T x - y^T(A^T x - b) + \frac{1}{2}\rho_k \|A^T x - b\|^2}$
    2. Find $\mathcal{A}(x_k^c)$ & estimate penalty $\bar{\rho} = 2\,\|H_{\mathcal{I}}\|/\|A_{\mathcal{I}}A_{\mathcal{I}}^T\|$
    3. IF $\bar{\rho} > \rho_k$ THEN update $\rho_{k+1} = \bar{\rho}$ & CYCLE
       ELSE update multiplier: $y_k^c = y_k - \rho_k(A^T x_k^c - b)$
    4. Solve equality QP in subspace $\rightarrow \; (\Delta_{x_{\mathcal{I}}}, \Delta y)$
       $\begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}} \\ A_{:,\mathcal{I}}^T & \end{bmatrix} \begin{pmatrix} \Delta x_{\mathcal{I}} \\ \Delta y \end{pmatrix} = - \begin{pmatrix} [\nabla x L(x_k^c, y_k^c, 0)]_{\mathcal{I}} \\ A^T x_k^c - b \end{pmatrix}$
    5. Line-search on $L(x_k^c + \alpha\Delta x, y_k^c + \alpha\Delta y, \rho)$; update $x, y, k, \rho$
END
```

1.-3. identify active set ... 4. gives fast convergence

Filter Methods

A Filter for Augmented Lagrangian Methods

Two competing aims in augmented Lagrangian:

1. reduce $h_k := \|A^T x_k - b\| \le \eta_k \searrow 0$
2. reduce $\theta_k := \|\nabla L(x_k, y_k, \rho_k) - z_k\| \le \omega_k \searrow 0$

A Filter for Augmented Lagrangian Methods

Two competing aims in augmented Lagrangian:

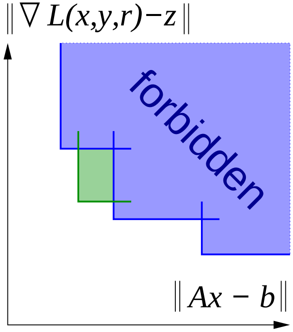
1. reduce $h_k := \|A^T x_k - b\| \leq \eta_k \searrow 0$
2. reduce $\theta_k := \|\nabla L(x_k, y_k, \rho_k) - z_k\| \leq \omega_k \searrow 0$

... why should one sequence $\{\omega_k\}, \{\eta_k\}$ fit *all problems ???*

A Filter for Augmented Lagrangian Methods

Introduce a filter $\mathcal{F}$ to promote convergence

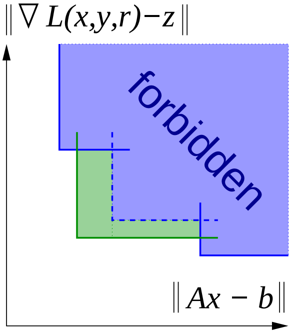
- list of pairs $(\|A^T x_l - b\|, \|\nabla L_l - z_l\|)$
- no pair dominates any other pair
- new x_k acceptable to filter $\mathcal{F}$, iff
    1. $h_k \leq 0.99 \cdot h_l \; \forall l \in \mathcal{F}$
    2. $\theta_k \leq 0.99 \cdot \theta_l \; \forall l \in \mathcal{F}$



A Filter for Augmented Lagrangian Methods

Introduce a filter $\mathcal{F}$ to promote convergence

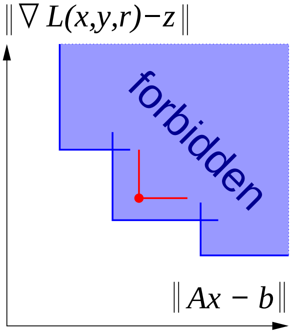
- list of pairs $(\|A^T x_l - b\|, \|\nabla L_l - z_l\|)$
- no pair dominates any other pair
- new x_k acceptable to filter $\mathcal{F}$, iff
  1. $h_k \leq 0.99 \cdot h_l \; \forall l \in \mathcal{F}$
  2. $\theta_k \leq 0.99 \cdot \theta_l \; \forall l \in \mathcal{F}$
- remove redundant entries



A Filter for Augmented Lagrangian Methods

Introduce a filter $\mathcal{F}$ to promote convergence

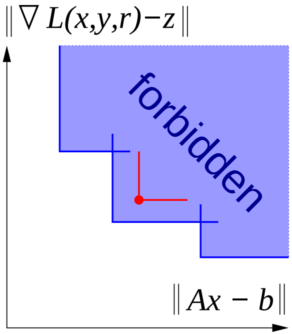
- list of pairs $(\|A^T x_l - b\|, \|\nabla L_l - z_l\|)$
- no pair dominates any other pair
- new x_k acceptable to filter $\mathcal{F}$, iff
    1. $h_k \leq 0.99 \cdot h_l \; \forall l \in \mathcal{F}$
    2. $\theta_k \leq 0.99 \cdot \theta_l \; \forall l \in \mathcal{F}$
- remove redundant entries
- reject new x_k, if $h_k \geq h_l$ & $\theta_k \geq \theta_l$



A Filter for Augmented Lagrangian Methods

Introduce a filter $\mathcal{F}$ to promote convergence

- list of pairs $(\|A^T x_l - b\|, \|\nabla L_l - z_l\|)$
- no pair dominates any other pair
- new x_k acceptable to filter $\mathcal{F}$, iff
  1. $h_k \leq 0.99 \cdot h_l \; \forall l \in \mathcal{F}$
  2. $\theta_k \leq 0.99 \cdot \theta_l \; \forall l \in \mathcal{F}$
- remove redundant entries
- reject new x_k, if $h_k \geq h_l$ & $\theta_k \geq \theta_l$



... and old friend from Chicago ...

Augmented Lagrangian Cauchy Pointe (Al Capone)

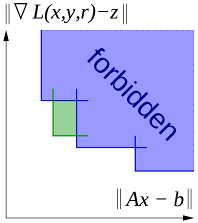
Requirement on Cauchy Point x_k^c for filter:

1. x_k^c, y_k^c acceptable to filter

Augmented Lagrangian Cauchy Pointe (Al Capone)

Requirement on Cauchy Point x_k^c for filter:

1. x_k^c, y_k^c acceptable to filter
2. $\|\nabla L(x_k, y_k, \rho_k) - z_k\| \leq \omega_k$
   ... optimality of Lagrangian



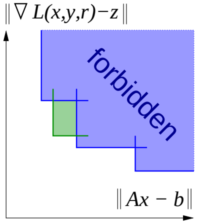
Augmented Lagrangian Cauchy Pointe (Al Capone)

Requirement on Cauchy Point x_k^c for filter:

1. x_k^c, y_k^c acceptable to filter
2. $\|\nabla L(x_k, y_k, \rho_k) - z_k\| \leq \omega_k$
   ... optimality of Lagrangian

   New: $\omega_k := 0.1 \max\{\|\nabla L_l - z_l\|\}$

   ... depends on filter



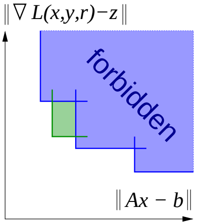
Augmented Lagrangian Cauchy Pointe (Al Capone)

Requirement on Cauchy Point x_k^c for filter:

1. x_k^c, y_k^c acceptable to filter
2. $\|\nabla L(x_k, y_k, \rho_k) - z_k\| \leq \omega_k$
   ... optimality of Lagrangian

   New: $\omega_k := 0.1 \max \{\|\nabla L_l - z_l\|\}$

   ... depends on filter



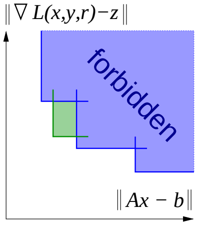
1. ensures that back-tracking line-search will succeed
   ... if not acceptable then reduce $\omega_{k+1} = \omega_k/2$

Augmented Lagrangian Cauchy Pointe (Al Capone)

Requirement on Cauchy Point x_k^c for filter:

1. x_k^c, y_k^c acceptable to filter
2. $\|\nabla L(x_k, y_k, \rho_k) - z_k\| \leq \omega_k$
   ... optimality of Lagrangian

   New: $\omega_k := 0.1 \max \{\|\nabla L_l - z_l\|\}$
   ... depends on filter



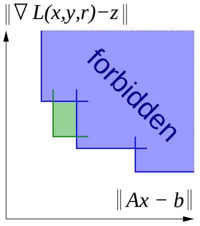
1. ensures that back-tracking line-search will succeed
   ... if not acceptable then reduce $\omega_{k+1} = \omega_k/2$
2. & $\omega_{k+1} = \omega_k/2$ ensure new entry can be added to filter

Augmented Lagrangian Cauchy Pointe (Al Capone)

Requirement on Cauchy Point x_k^c for filter:

1. x_k^c, y_k^c acceptable to filter
2. $\|\nabla L(x_k, y_k, \rho_k) - z_k\| \leq \omega_k$
   ... optimality of Lagrangian

   New: $\omega_k := 0.1 \max\{\|\nabla L_l - z_l\|\}$
   ... depends on filter

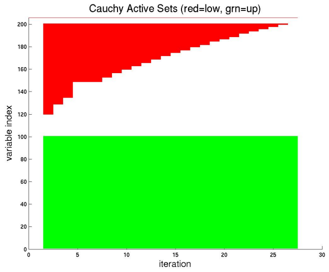


1. ensures that back-tracking line-search will succeed
   ... if not acceptable then reduce $\omega_{k+1} = \omega_k/2$
2. & $\omega_{k+1} = \omega_k/2$ ensure new entry can be added to filter

Why do you keep the penalty parameter?
... combines search directions for $\|A^T x - b\|$, and
$\|\nabla L(x_l, y_l, \rho_l) - z_l\|$
$\Rightarrow$ gradient projection possible

Active Set Evolution: `blockqp4_100`

AUGLAG

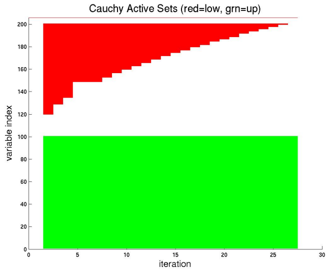


red = lower bound active

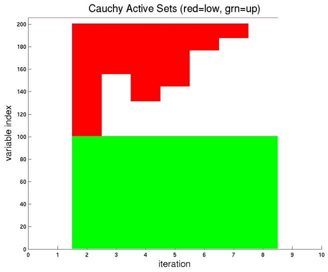
green = upper bound active

Active Set Evolution: `blockqp4_100`

AUGLAG



FILTER



red = lower bound active

green = upper bound active

Summary: Active Set QP Method

1. **Global Convergence**

2. **Active Set Identification**

3. **Fast Local Convergence**

Summary: Active Set QP Method

1. **Global Convergence**
   - augmented Lagrangian & filter
     $\Rightarrow$ no arbitrary parameters
2. **Active Set Identification**
3. **Fast Local Convergence**

Summary: Active Set QP Method

1. **Global Convergence**
   - augmented Lagrangian & filter
     $\Rightarrow$ no arbitrary parameters
2. **Active Set Identification**
   - projected gradient on augmented Lagrangian
   - easy penalty parameter estimate
3. **Fast Local Convergence**

Summary: Active Set QP Method

1. **Global Convergence**
   - augmented Lagrangian & filter
     $\Rightarrow$ no arbitrary parameters
2. **Active Set Identification**
   - projected gradient on augmented Lagrangian
   - easy penalty parameter estimate
3. **Fast Local Convergence**
   - conjugate gradients on equality QP
   - (constrained) preconditioners ???
   - Benzi-Golub ... ties in with augmented Lagrangian

ACTIVE SET METHODS FOR NLPs

Sequential Quadratic Programming (SQP)

$$\text{NLP:} \qquad \underset{x}{\text{minimize}} \; f(x) \qquad \text{subject to } \; c(x) \geq 0$$

SQP method of choice for NLP

Compute displacement/step d by solving QP subproblem

$$\boxed{\begin{array}{lll} \underset{d}{\text{minimize}} & g^T d + \frac{1}{2} d^T W d & \\ \text{subject to} & c + A^T d \geq 0 & \\ & \|d\|_\infty \leq \Delta & \text{Trust-Region} \end{array}}$$

where $g = \nabla f(x)$, $A = \nabla c(x)^T$, $W = \nabla^2 \mathcal{L}(x, y)$

Sequential Quadratic Programming (SQP)

```
WHILE (not optimal) BEGIN
  1. Compute displacement/step $d$ by solving QP subproblem
  2. IF step $d$ acceptable THEN
           $x = x + d$ & increase trust-region radius $\Delta = 2 * \Delta$
     ELSE
           $x = x$ & decrease trust-region radius $\Delta = \Delta/2$
END
```

- How to make it work for *n* large ???
  QP solve is bottleneck ... could use new QPFIL
- $\exists$ excellent LP solvers ... but QP harder

Sequential Linear Programming

Throw away quadratic term $\Rightarrow$ linear program

Compute displacement/step d_{LP} by solving LP subproblem

$$\boxed{\begin{array}{lll} \underset{d}{\text{minimize}} & g^T d + \frac{1}{2} d^T W d & \\ \text{subject to} & c + A^T d \geq 0 & \\ & \|d\|_\infty \leq \Delta & \text{Trust-Region} \end{array}}$$

where $g = \nabla f(x)$, $A = \nabla c(x)^T$, $W = \nabla^2 \mathcal{L}(x, y)$

Sequential Linear Programming

```
WHILE (not optimal) BEGIN
  1. Compute displacement/step $d_{LP}$ by solving LP subproblem

  3. IF step $d$ acceptable THEN
         $x = x + d$ & increase trust-region radius $\Delta = 2 * \Delta$
     ELSE
         $x = x$ & decrease trust-region radius $\Delta = \Delta/2$
END
```

$\Rightarrow$ slow local convergence ... steepest descent

Sequential Linear Programming

```
WHILE (not optimal) BEGIN
    1. Compute displacement/step $d_{LP}$ by solving LP subproblem
    2. Identify active constraints: $\mathcal{A} = \{i : c_i + a_i^T d_{LP} = 0\}$ & solve
```

$$\begin{array}{lll} \underset{d}{\text{minimize}} & g^T d + \frac{1}{2} d^T W d & \\ \text{subject to} & c_i + a_i^T d = 0 & i \in \mathcal{A} \end{array} \Leftrightarrow \begin{bmatrix} H_{\mathcal{I},\mathcal{I}} & -A_{:,\mathcal{I}}^T \\ A_{:,\mathcal{I}} & \end{bmatrix} \left(\begin{array}{c} d_{\mathcal{I}} \\ y \end{array} \right.$$

```
       equality QP for step $d$
    3. IF step $d$ acceptable THEN
            $x = x + d$ & increase trust-region radius $\Delta = 2 * \Delta$
       ELSE
            $x = x$ & decrease trust-region radius $\Delta = \Delta/2$
END
```

How expensive are LPs??

Active Set Identification by SLP

Polyhedral trust-region makes LP solves inefficient

$$\boxed{\begin{array}{lll} \underset{d}{\text{minimize}} & g^T d & \\ \text{subject to} & c + A^T d \ge 0 & \\ & \|d\|_\infty \le \Delta & \text{Trust-Region} \end{array}}$$

- many changes to active trust-region bounds
- LP solvers too slow near solution

Active Set Identification by SLP

Ellipsoidal trust-region makes LPs into NLPs

$$\boxed{\begin{array}{lll} \underset{d}{\text{minimize}} & g^T d & \\ \text{subject to} & c + A^T d \geq 0 & \\ & \|d\|_2 \leq \Delta & \text{Trust-Region} \end{array}}$$

- trust-region (always) active $\Rightarrow$ no changes
- subproblem is now NLP ... as hard as original problem ???

Active Set Identification by SLP

Ellipsoidal trust-region makes LPs into NLPs

$$\boxed{\begin{array}{lll} \underset{d}{\text{minimize}} & g^T d & \\ \text{subject to} & c + A^T d \geq 0 & \\ & d^T d \leq \Delta^2 & \text{Trust-Region} \end{array}}$$

- trust-region (always) active $\Rightarrow$ no changes
- subproblem is now NLP ... as hard as original problem ???

Summary: Active Set NLP Method

1. **Global Convergence**

2. **Active Set Identification**

3. **Fast Local Convergence**

Summary: Active Set NLP Method

1. **Global Convergence**
   - Sequential bound constraint
   - Trust-region & Filter
2. **Active Set Identification**
3. **Fast Local Convergence**

Summary: Active Set NLP Method

1. **Global Convergence**
    - Sequential bound constraint
    - Trust-region & Filter
2. **Active Set Identification**
    - d_{LP} LP steps from subproblem
3. **Fast Local Convergence**

Summary: Active Set NLP Method

1. **Global Convergence**
   - Sequential bound constraint
   - Trust-region & Filter
2. **Active Set Identification**
   - d_{LP} LP steps from subproblem
3. **Fast Local Convergence**
   - conjugate gradients on equality QP
   - (constrained) preconditioners ???